IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 7, 2019, accepted June 13, 2019, date of publication June 17, 2019, date of current version July 17, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2923530

Data Augmentation and Second-Order Pooling for Facial Expression Recognition

XIAOYUN TONG, SONGLIN SUN, (Senior Member, IEEE), AND MEIXIA FU

National Engineering Laboratory for Mobile Network Security, Beijing University of Posts and Telecommunications, Beijing 100876, China
Key Laboratory of Trustworthy Distributed Computing and Service (BUPT), Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China
School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Xiaoyun Tong (email: xiaoyun_t@bupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Project 61471066, and in part by the Open Project Fund of the National Key Laboratory of Electromagnetic Environment, China, under Grant 201600017.

ABSTRACT Facial expression is the main medium of information transmission in human communication, playing an important role in human's daily life. Facial expression recognition is still challenging due to the various obstacle, illumination, and posture. However, most of the existing works focus on deeper or wider network structures and rarely explores the high-level feature statistics. In this paper, we propose a second-order pooling convolution neural network to explore the correlation information between the facial features after deep network learning. At the final stage of the network, we add a new covariance pooling layer to replace the first-order pooling of standard convolution networks. In the pooling layer of covariance, the Newton iteration method is used to approximate the square root instead of EIG or SVD, which makes it more suitable for GPU. Due to the small amount of facial expression data, this paper uses different data augmentation methods to increase the amount of training data and improve the generalization ability of the model. The proposed method, data augmentation and second-order pooling (DASOP), was evaluated on the real-world affective faces database (RAFDB) and the static facial expressions in the wild (SFEW), yielding correct rates of 88.625% and 59.518%, respectively. We achieve state-of-the-art performance superior to existing methods.

INDEX TERMS Facial expression recognition, data augmentation, second-order pooling, deep convolutional neural networks.

I. INTRODUCTION

Facial expression is the most natural and direct way to reflect people's inner emotions and thoughts. Therefore, the research of facial expression recognition technology based on visual information is another research topic after face detection and recognition [1]. Facial expression recognition has potential application value in many fields, such as intelligent medical care, intelligent monitoring in security field, human-computer interaction, digital entertainment, etc. In facial expression recognition, it is a very challenging task because the visual differences between different subclasses are small and easily affected by posture, perspective, illumination, and so on [2].

Deep convolutional neural networks generally require a large amount of training data to get ideal results. However, the amount of public database of facial expression recognition is limited, so data augmentation can be used to increase the diversity of training samples, improve the robustness and generalization ability of the model, and avoid over-fitting [3]. For example, we can crop the image in different ways so that the objects appear at different positions in the image at different scales, which can reduce the sensitivity of the model to the target position.

Convolutional neural network (CNNs) have enough capacity to represent the complex variations of samples by learning features hierarchically [4]. Therefore, CNNs have made outstanding achievements in facial expression recognition. The common models mainly adopt the average or max pooling layer in designing network architectures. However, these layers only capture the first-order statistics of input features, which limit the learning capacity of models [5]. In addition, the structure of face not only contains the global discriminant information, but also the local discriminant information.

The associate editor coordinating the review of this manuscript and approving it for publication was Abdullah Iliyasu.

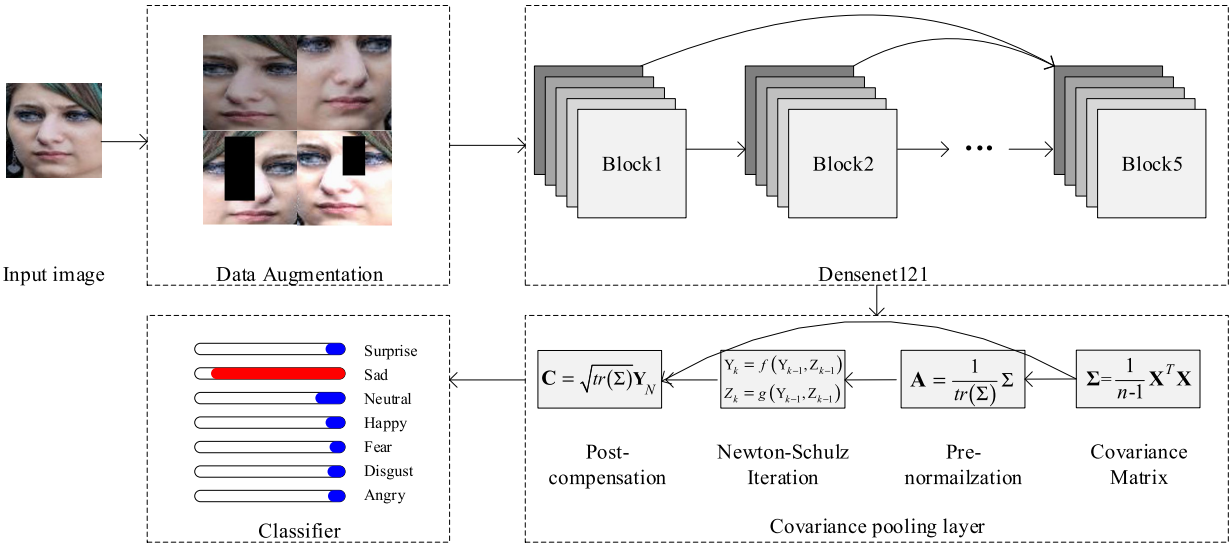

FIGURE 1. Data augmentation and second-order pooling based Densenet121 networks for facial expression recognition.

TABLE 1. Comparison of accuracies with various standard network architectures on RAFDB and SFEW.

Datasets	Methods	Original	Aug	DASOP
RAFDB	Alexnet	76.662	83.051	83.768
	Vgg16	80.900	85.332	86.147
	Inception_V3	71.741	84.746	86.669
	Resnet50	78.162	79.140	85.235
	Densenet121	78.651	85.137	**88.625**
SFEW	Alexnet	20.241	52.289	53.012
	Vgg16	20.241	53.012	58.798
	Inception_V3	27.711	50.843	53.494
	Resnet50	30.361	48.675	54.217
	Densenet121	38.313	54.217	**59.518**

So, the performance of single model is limited to the loss of discriminant information.

At present, deep facial expression recognition faces two key problems: (1) over-fitting due to lack of sufficient training data; (2) facial expression changes are subtle and changeable, the first-order information is insufficient to provide more discriminant information. To overcome these issues, this paper proposes data augmentation and second-order pooling based deep convolutional neural networks for facial expression recognition. An overview of the proposed methods is shown in Figure 1. The input image is transformed into a Densenet121 neural network, then the second-order information of the output feature is calculated, and the final output category is obtained. In summary, the main contribution of our work consists of:

(1) We add multiple data augmentation technologies to datasets, and set a group of experiments on different networks to illustrate the effectiveness of data augmentation. The result of experiments is shown in Table 1.

(2) Facial expression recognition adopts end-to-end pooling of second-order statistics. Newton iteration method is used to solve the covariance matrix to make it more suitable for GPU.

(3) The proposed DASOP achieves the state-of-art performance on two public datasets: RAFDB and SFEW.

The remainder of this paper is introduced as follows. Section II summarize the work most relevant to our work. Section III descripts the proposed model, including data augmentation, and covariance descriptors. The improved performance of the proposed DASOP on two public datasets, and comparison with the state-of-art solutions is demonstrated in Section IV. Finally, Section V concludes this paper and gives the directions of future work.

II. RELATED WORK

The proposed method DASOP consists of two parts: data augmentation and second-order pooling. Therefore, this paper introduces the related work of data augmentation and the second-order pooling separately, and summarizes as follows.

A. DATA AUGMENTATION

Convolutional neural networks can classify objects placed in different directions, or different scales, or different illuminations robustly, that is, convolutional neural networks have invariant properties [3]. This is the premise of data augmentation.

Data augmentation is mainly to improve the robustness and generalization ability of the model, and reduce the over-fitting phenomenon of the network. By transforming the training data, the network with stronger generalization ability can be obtained, which can better adapt to the application scenarios. Over-fitting is mainly caused by two reasons: too little data and too complex model. Hence, data augmentation is the most direct and effective method to avoid over-fitting.

Commonly data augmentation methods listed as follows:

Rotation: Random rotates the image at a certain angle.

Reflection: change the orientation of the image content.

Flip: Flip the image horizontally or vertically.

Scale: The image can be outward or inward according to the specified scale factor; or the scale space is constructed by filtering the image with using the specified scale factor, according to the idea of SIFT feature extraction.

Crop: Random select the region of the image for clipping and scaling to a certain scale.

Shift: shifts the image in a certain way on the image plane.

Noise: Random perturbation of each pixel of the image. The common noise modes are salt and pepper noise and Gauss noise.

Color jitter: Randomly change the brightness, contrast and saturation of the image.

Random Erasing [6]**:** Random selection of an area on a picture and random erasure of image information.

Generative Adversarial Networks (GAN) [7]**:** An image generation model consists of two parts, a generator and a discriminator, which generate images through two-part mutual game learning.

In order to get more data, we just need to make minor changes to our existing dataset, such as flips, translations or rotations. Data augmented images are distinct and independent images for untrained neural networks.

B. SECOND-ORDER POOLING

Deep convolutional neural networks have achieved great success in large-scale object classification and other areas of computer vision. In fact, this network can be seen as a process of learning low level features through hierarchical convolution and pooling, and getting high level representation through a global average pooling that is sent it to the classifier for classification.

The focus here is global average pooling. This layer of work was first proposed on [8], and is now widely used in mainstream deep networks, such as Inception, Resnet, Densenet and so on. However, the problem of global average pooling is that the previous network acquires a strong expressive feature through continuous learning, but at last it makes a global mean when it expresses the image. Statistically speaking, the mean is only a first-order information that is often simple, fast and effective. However, it is easy to ignore the relevant information between channels by calculating the mean of each channel. Based on this point, many authors propose a second-order statistical method to replace the first-order global average pooling, and thus make a series of work.

One of the earliest works employing second-order pooling for semantic segmentation used it as regional descriptor, combined with LBP, SIFT and MSIFT [9]. Previous well-known work included DeepO2P [5] and Bilinear Model [10] and various improvements. In [11], authors propose a new class of CNN architectures to employ covariance pooling. The method of global covariance pooling is first applied to large-scale visual recognition tasks in [12]. It not only given some theoretical analysis, but also prove to be effective in large-scale vision problems, such as the classification of ImageNet. In [13], authors propose symmetric positive definite manifold network, that is, covariance pool is used in facial expression recognition, combining manifold networks with traditional convolutional networks, and using spatial pools in a single image feature map in an end-to-end depth learning approach. The following is a brief description of the covariance processing for different second-order pooling.

Deep Second Order Pooling (DeepO2P) [5]: adopts Log-Euclidean (Log-E) metric for exploiting geometry of covariance spaces, which only brings side effect on covariance representations. It is not suitable for large-scale datasets, such as ImageNet.

Bilinear Pooling (BP) [10]: performs element-wise square root normalization, without considering the manifold of covariance matrices. Bilinear convergence calculates the outer product of different spatial locations and calculates the average convergence for different spatial locations to obtain bilinear features. The outer product captures the pairwise correlation between the feature channels, and this is translationally invariant. Bilinear convergence provides a stronger representation of the feature than the linear model, and can be optimized end-to-end to achieve performance.

Compact Bilinear Pooling (CBP) [14]: use the outer product to combine the feature maps of the two stream CNNs, then transform the bilinear features through the square root of the symbol, and add L2 normalization, input classifier. Compared with the BP, the loss of the CBP is basically unchanged, but the size of the parameters is reduced by two orders of magnitude, and CBP supports an end-to-end training structure.

Global Gaussian Distribution Embedding Network (G2DeNet) [15]: take advantage of matrix normalization but suffer from large dimensionality since they use the square-root of a large pooled matrix. At the same time, the authors use the first and second-order information and normalize it by 0.5 matrix power.

Moments Embedding Network (MoNet) [16]: using a sub-matrix square root layer, matrix normalization is performed before bilinear pooling, combined with compact pooling, reduces dimensionality significantly without compromising performance. The description matrix is augmented, which contains both first and second-order information. In addition, tensor sketch can be used to design a compact bilinear merge operation. Its performance is better than G2DeNet.

Symmetric Positive Definite Manifold Network (SPDNET) [13]: covariance features can be flattened through Bilinear Mapping Layer, and then introduced non-linearity by Eigenvalue Rectification, Log Eigenvalue Layer endows elements in Riemannian manifold with a Lie Group structure.

Matrix Power Normalization [12]: the covariance matrix of depth description vector is normalized by 0.5 matrix power to obtain bilinear convergence feature. The covariance is by

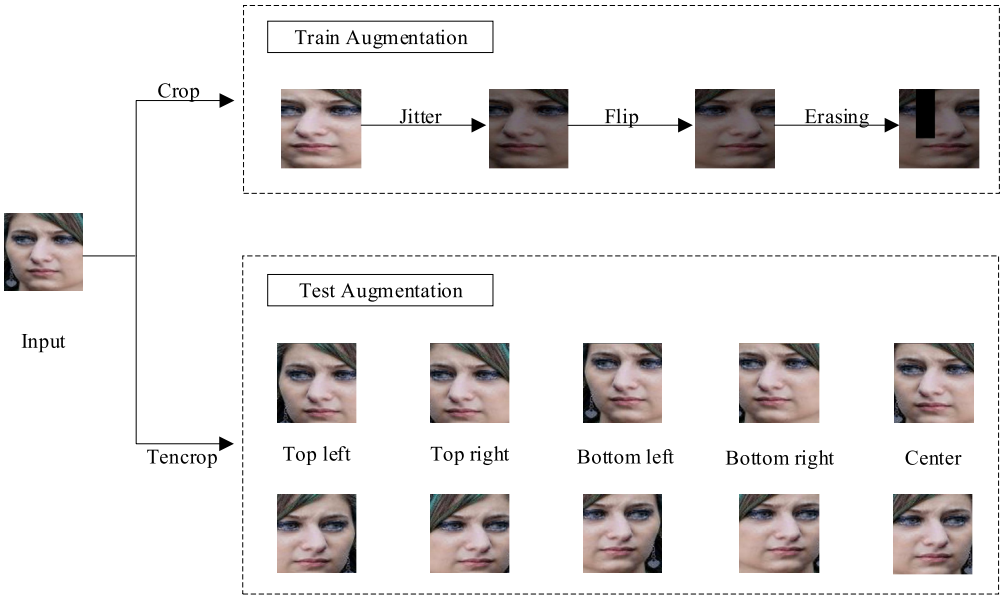

FIGURE 2. Data augmentation in train and test datasets.

eigen-decomposed to obtain the eigenvalue, and then the square root of eigenvalue is obtained. It solves the problem of high dimension statistics of small samples, and effectively uses the geometric structure of covariance matrix [17]. However, almost all platforms based on GPU have poor support for eigen decomposition, so its efficiency is very poor.

Iterative Matrix Square Root Normalization of Covariance Pooling [18]: because there is no efficient implementation of eigenvalue decomposition and SVD on GPU, Schur decomposition or eigenvalue decomposition are still needed when Lyapunov equation is solved in reverse. It is proposed that both forward and backward processes are based on Newton iteration method. The matrix is normalized by 0.5 matrix power.

A common framework of above references is to use CNN to extract feature map, and then use each location in feature map as a sample to calculate the covariance matrix of the entire feature map. After the covariance matrix is obtained, the eigenvalues are reduced, parameterized or normalized and then output.

Facial expression recognition is more directly related to the deformation of facial feature points than to the existence or absence of specific feature points. Therefore, we believe that if geometric moments are applied to facial expression recognition, the second-order statistics are more suitable than other statistics for capturing the regional distortion of key points in facial features.

In the generated convolution network, global covariance pooling is a better choice than global average pooling. There are three reasons: (1) better performance and stronger generalization ability; (2) good theoretical explanation in statistics and geometry; (3) fast convergence and high computational efficiency.

III. PROPOSED METHOD

Facial expression recognition mainly includes three parts: face detection, feature extraction, expression recognition. That is, given a picture, face detection is performed first, and then align based on facial markers. The extracted face is augmented and standardized, and then fed into the deep neural network. To consider the interaction between features, we use second-order pooling, and then employ the Newton-Schulz iteration to explore second-order statistical information. Because the difference between classes is not significant, the first-order information is not applicable, but the second-order information can bring more discriminatory and valuable information to the classifier.

A. DATA AUGMENTATION

Because of too little data and too many network parameters, it is easy to lead to over-fitting. The facial expressions publicity dataset is small, and the categories are not balanced, so this paper adopts data augmentation methods to expand the dataset to avoid over-fitting. Another advantage of data augmentation is to expand the amount of data in the database, which makes the training network more robust. The data augmentation of training used in this paper are as follows: resize, random horizontal flip, random crop and random Erasing. Random erasing is complementary to random clipping and random horizontal flipping. The testing data is cropping the given image four corners and the central crop plus the flipped version of these, then average the predictions of 10 images, the largest output classification is corresponding expression. This method effectively reduces classification errors. Data augmentation is shown in Figure 2.

B. COVARIANCE POOLING

After deep convolutional neural network, feature will be obtained. Firstly, covariance matrix is calculated. Unlike first-order pooling, covariance matrix can describe the correlation between each channel.

Defining the output of the convolutional network is a tensor of $h \times w \times d$ dimensions, where h is the height, w is the width, and d is the number of channels. This tensor is reshaped into a $n \times d$ dimensional feature matrix, where $n = h \times w$. Then calculate the covariance of this matrix:

$$\mathbf{\Sigma} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T = \frac{1}{n-1}\mathbf{X}^T\mathbf{X} \quad (1)$$

where $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ be the set of features, $\mathbf{X}$ denotes the mean-subtracted data matrix.

At present, almost all GPU-based platforms have very poor support for eigenvalue decomposition, so the second-order pooling layer in this paper uses iterative method to solve the square root of covariance. Since the iteration itself is not global convergence, it can be guaranteed by dividing the trace of the matrix by the covariance matrix; however, the covariance is changed after dividing by the trace, so the value of the trace is compensated by post-compensation after the iteration. Detailed description is as follows.

1) PRE-NORMALIZATION

In order to ensure the global convergence of the proposed algorithm, we pre-normalize covariance matrix by dividing its trace before the Newton iteration, i.e.,

$$\mathbf{A} = \frac{1}{tr\,(\Sigma)}\Sigma \quad (2)$$

Let λ_i be eigenvalues of Σ, arranged in nondecreasing order. As $tr\,(\Sigma) = \sum_i \lambda_i$, then:

$$\|\mathbf{\Sigma} - \mathrm{I}\|_2 = \max(\mathbf{\Sigma} - \mathrm{I}) = 1 - \frac{\lambda_1}{\sum_i \lambda_i} < 1 \quad (3)$$

Therefore, the proposed algorithm satisfies the convergence condition.

2) NEWTON-SCHULZ ITERATION

The idea of Newton's method is simple. Given an initial point, the tangent at that point is used to approximate the function, and then the root of the tangent is found as an iteration. Newton's iteration method is to make the equation converge gradually through continuous iteration.

Let $\mathbf{A}$ be an SPD matrix, the square root of $\mathbf{A}$ is $\mathbf{Y} = \mathbf{U}diag\left(\lambda_i^{1/2}\right)\mathbf{U}^T$, but both EIG and SVD are not well supported on GPU [12]. Hence, we computing $\mathbf{Y}$ by Newton Schulz iteration, a case of coupled iteration as follows:

$$\begin{aligned}\mathbf{Y}_k &= \frac{1}{2}\mathbf{Y}_{k-1}(3\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Y}_{k-1}) \\ \mathbf{Z}_k &= \frac{1}{2}(3\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Y}_{k-1})\mathbf{Z}_{k-1}\end{aligned} \quad (4)$$

where $\mathbf{Z} = \mathbf{I}$ for k $= 1, \ldots, N$.

Equation (4) only involves matrix product operations, not matrix inverse, hence is suitable for implementations on GPUs. Compared to the exact square root of the EIG calculation, the approximate solution can only be obtained by 5 iterations [13].

The reason for exponentiation is to solve the problem of small sample high dimension in covariance estimation. Reference [8] has been verified that when the power is 0.5, that is, the square root operation, the effect is optimal.

3) POST-COMPENSATION

Since the pre-normalization changes the covariance size, in order to eliminate this effect, post-compensation is added after the Newton iteration method to compensate the trace value back, i.e., $\mathbf{C} = \sqrt{tr\,(\Sigma)}\mathbf{Y}_N$

Finally, the output of the second-order pooling layer is fed into the classifier for classification.

IV. EXPERIMENTS

In this section, we conduct many experiments of the proposed method on two public datasets and present the state-of-the-art performance though comparing with the previous work. The proposed system is implemented on Pytorch using NVIDIA GeForce GTX 1080Ti with 11GB memory. In our experiments, the input image is 224∗224, the batch size is 32. The training epoch is set to 100. The initial learning rate is 0.1 and decays by a factor of 0.1 every 30 epochs. DNNs adopted in Table 1 is trained from scratch.

A. DATASETS

In this paper, facial expression datasets in the wild are chosen, which can better approximation to the real-world scenarios, and also more challenging. We present the evaluation results of RAFDB and SFEW datasets, and reveal the relevance and importance of facial features.

Real-world Affective Faces Database (RAFDB) [19] is a large-scale facial expression database with around 30K great-diverse facial images downloaded from the Internet. This dataset contains 15331 images and that have been labeled for seven basic expressions. There are 12271 samples in training set and 3068 samples in test set. The RAF-DB is 100∗100 RGB image, is shown in figure 3(a). There are 7 rows, each row representing a category with 7 images per category. Figure 3(b) is the data distribution. It shows the number of images per emotion in the training set and test set. As can be seen from Figure 3(b), the data distribution is very uneven.

SFEW [20] extracted from a temporal facial expressions database Acted Facial Expressions in the Wild which we have extracted from movies. The database covers unconstrained facial expressions, varied head poses, large age range, occlusions, varied focus, different resolution of face and close to real world illumination. In total, SFEW contains 1766 images and that have been labeled for seven basic expressions. There are 958 samples in training set and 436 samples in validation set. Since the test set are not labeled, here the results on the validation set is reported in this paper.

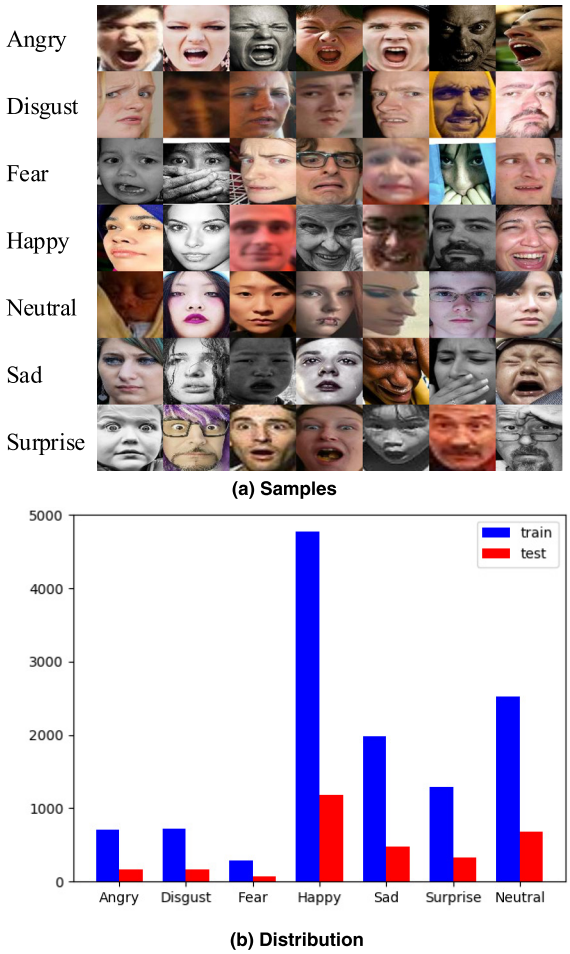

FIGURE 3. RAFDB samples and distribution.

B. RESULTS AND DISCUSSION

Since the images provided by the dataset contain many backgrounds, it needs to be preprocessed, that is, face extraction. For SFEW, we use the MTCNN proposed in [21] to detect faces. For RAFDB, the dataset provides the aligned faces.

We evaluate five current classical CNN architecture as backbone network for DASOP in Table 1. we present the comparison of accuracies with various standard network architectures on RAFDB and SFEW. Original means the original data input to different networks for training. Aug is for data augmentation using standard techniques including random crop, random flip and random erasing. DASOP is data augmentation and covariance pooling.

It can be seen from Table 1 that:

(1) the densenet121 + DASOP of method proposed in this paper obtains best classification results with accuracy 88.625% on RAFDB and 59.518% on SFEW. This shows that the deep convolution network has a very obvious effect on the classification. The corresponding confusion matrixes are shown in Figure 3.

(2) Data augmentation is very effective to improve the classification effect. The accuracy rate is increased by 1-7% on RAFDB and 16-32% on SFEW, respectively. The SFEW datasets are fewer, so the improvement of the accuracy rate is more obvious.

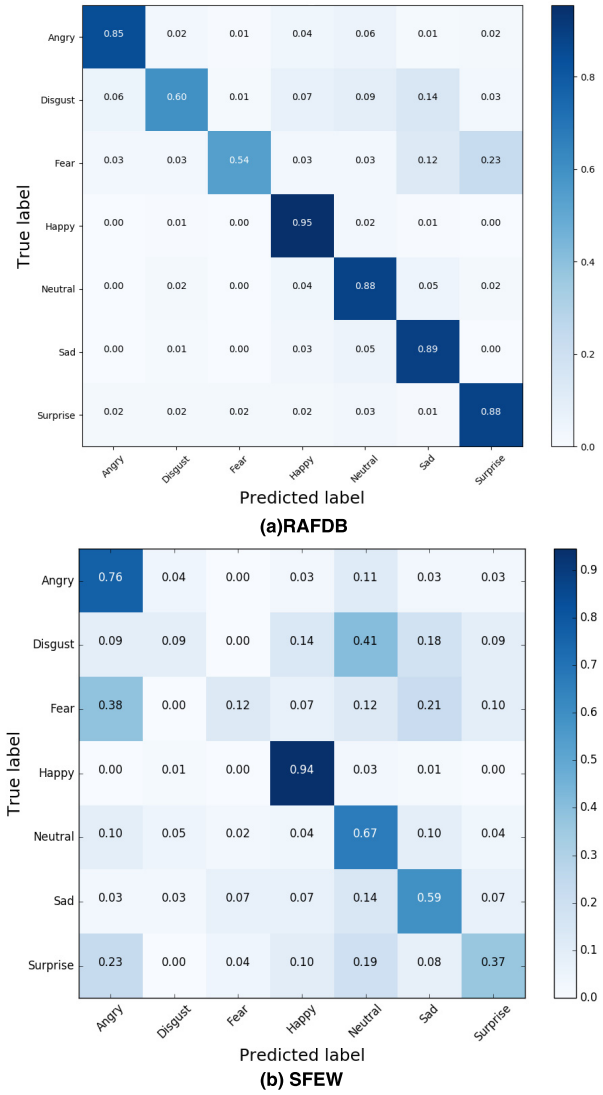

FIGURE 4. Confusion matrix for Densenet121 and DASOP on the RAFDB and SFEW.

(3) Compared with the first-order pooling, the accuracy of the second-order pooling is improved by 0.7-6% on RAFDB and 1-6% on SFEW, respectively, which shows the effectiveness of the second-order pooling and provides more useful information.

Figure 4 shows that the accuracy of happiness and angry is significantly higher than others, but the accuracy of fear and disgust is very low. For this problem, we think that the first reason is that the number of different expressions in the data set is not balanced, this kind of imbalance is enough to make the classification error. The second reason is that surprise, disgust, fear and sad have certain similarities. In real life, people also find it difficult to distinguish these four kinds of expressions, especially when they are not acquainted with

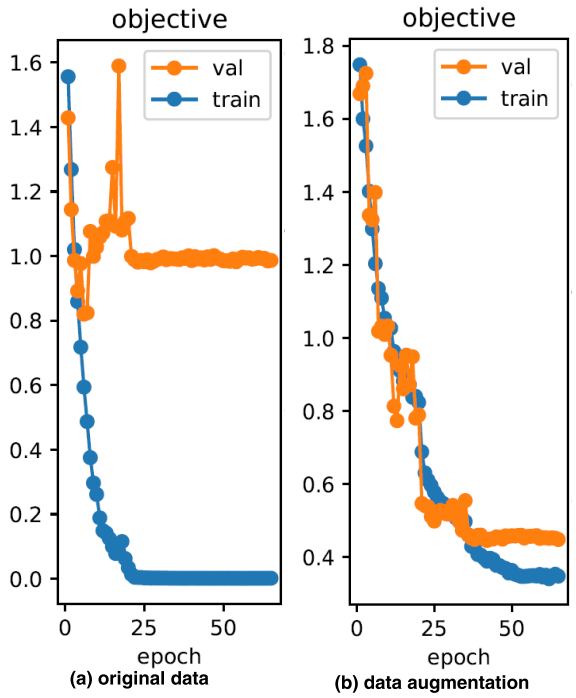

FIGURE 5. The loss function curve for original data and data augmentation with Densenet121 on the RAFDB.

each other, it is even more difficult to correctly recognize expressions. Moreover, we find that misjudgments always occur in some classes, which may be really difficult to distinguish and confuse.

In RAFDB, the loss function curve of original data and data augmentation based on densnet121 is shown in Figure 5. From Figure 5 (a), it can be seen that there is a serious over-fitting problem in the original data, so data augmentation technology is used to avoid over-fitting in this paper. Figure 5 (b) shows that data augmentation can solve this problem very well.

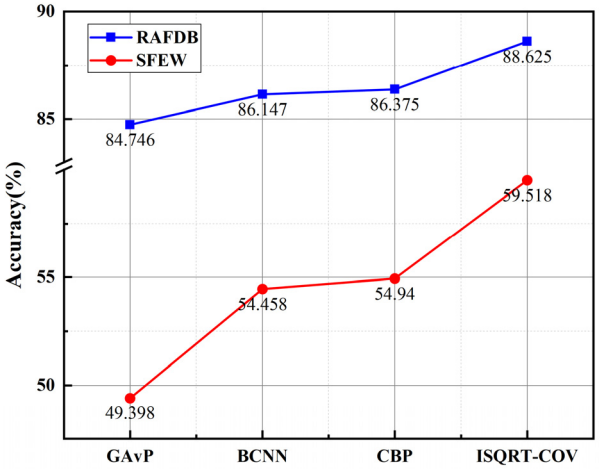

FIGURE 6. comparison of densenet121 + DASOP with other pooling.

C. COMPARE TO OTHER POOLING

In this section, we evaluate different second-order pooling methods for facial expression recognition. Figure 6 shows the comparison of the proposed method with the first-order global average pooling (GAvP) method and the second-order pooling methods, BP with CNN (BCNN) and CBP. According to the Table 1 experimental results, Densenet121 was chosen as the basic network. As can be seen from Figure 6, the result of second-order pooling is higher than first-order pooling with accuracy 2-4% on RAFDB and 5-10% on SFEW, respectively. And our method achieves the highest accuracy in second-order pooling methods.

D. COMPARE TO OTHER REFERENCE

Table 2 shows the comparison of the accuracy of each class between this paper and other references on RAFDB and SFEW. It can be seen that our algorithm is 1.625% higher than the current highest result, achieves the state-of-art results.

TABLE 2. The comparison of the accuracy of each class between this paper and other references on RAFDB and SFEW.

Datasets	Methods	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise	Average
RAFDB	Li et al. [19]	0.72	0.52	0.62	0.93	0.80	0.81	0.80	0.74
	Fan et al. [22]	0.84	0.58	0.61	0.89	0.80	0.86	0.80	0.77
	Li et.al [23]	-	-	-	-	-	-	-	0.85
	Zeng et al. [24]	-	-	-	-	-	-	-	0.87
	Acharya et al. [13]	0.80	0.61	0.61	0.93	0.89	0.86	0.86	0.87
	DASOP (ours)	0.85	0.60	0.54	0.95	0.88	0.89	0.88	**0.89**
SFEW	Yu et al. [25]	0.61	0.04	0.06	0.88	0.58	0.39	0.76	0.56
	Liu et al. [26]	0.66	0.04	0.06	0.88	0.58	0.40	0.73	0.54
	Li et al. [19]	-	-	-	-	-	-	-	0.51
	Li et al. [23]	-	-	-	-	-	-	-	0.53
	Acharya et al. [13]	0.66	0.00	0.14	0.90	0.86	0.66	0.29	0.58
	DASOP (ours)	0.76	0.09	0.12	0.94	0.67	0.59	0.37	**0.60**

This verifies the effective and significant of the proposed method. From all results based-deep learning, we can obtain that this stems from the effectiveness of deep convolutional networks and second-order information for feature extraction.

V. CONCLUSION

The novelty of the proposed model is that second-order statistical information is introduced into the network as image representation. Compared with the classical method, only the first-order statistical information is mined in the learning process, and the visual features with stronger resolving power can be learned. By introducing global covariance aggregation and matrix power normalization techniques, the proposed model is significantly better than the classical convolutional network in performance, and the convergence speed is faster. Data augmentation reduces the risk of over-fitting and makes the model robust to occlusion. Experiment conducted on RAFDB and fer2013 with various architectures validate the effectiveness of our method.

FUTURE WORKS

In this work, we use traditional supervised methods for data enhancement. However, in the real world, data in the natural state exists in a variety of situations, and cannot be handled by the above simple methods. Therefore, data enhancement can be achieved by the following unsupervised methods:

(1) Through the distribution of model learning data, the images which are consistent with the distribution of training data set are randomly generated. The representative method, GAN.

(2) Through the model, learn the data enhancement method suitable for the current task, the representative method, AutoAugment [27].

Also, we can add Global Second-order Pooling [28] across from lower to higher layers of deep convolutional networks aiming to learn more discriminative representations by exploiting the second-order statistics of holistic image throughout a deep convolutional network. This will be an interesting direction, and the results should be improved in theory.

REFERENCES

[1] S. Li and W. Deng, "Deep facial expression recognition: A survey," 2018, *arXiv:1804.08348*. [Online]. Available: https://arxiv.org/abs/1804.08348
[2] Y. Zong, X. Huang, W. Zheng, Z. Cui, and G. Zhao,, "Learning from hierarchical spatiotemporal descriptors for micro-expression recognition," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3160–3172, Nov. 2018.
[3] K. He, X. Zhang, S. Ren, and J. Sun "Deep residual learning for image recognition," in *Proc. CVPR*, Jun. 2016, pp. 770–778.
[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1–9.
[5] C. Ionescu, O. Vantzos, and C. Sminchisescu, "Matrix backpropagation for deep networks with structured layers," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2965–2973.
[6] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," 2017, *arXiv:1708.04896*. [Online]. Available: https://arxiv.org/abs/1708.04896
[7] I. Goodfellow, J. Pouget-Abadie, and M. Mirza, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[8] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proc. ICLR*, Mar. 2014, pp. 1–10.
[9] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu, "Semantic segmentation with second-order pooling," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 430–443.
[10] T. Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1449–1457.
[11] K. Yu and M. Salzmann, "Second-order convolutional neural networks," *Clin. Immunol. Immunopathol.*, vol. 66, no. 3, pp. 230–238, 2017.
[12] P. Li, J. Xie, Q. Wang, and W. Zuo, "Is second-order information helpful for large-scale visual recognition?" in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2070–2078.
[13] D. Acharya, Z. Huang, and D. Pani-Paudel, "Covariance pooling for facial expression recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2018, pp. 367–374.
[14] Y. Gao, O. Beijbom, and N. Zhang, "Compact bilinear pooling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 317–326.
[15] Q. Wang, P. Li, and L. Zhang, "G2DeNet: Global Gaussian distribution embedding network and its application to visual recognition," in *Proc. CVPR*, Jul. 2017, pp. 6507–6516.
[16] M. Gou, F. Xiong, and O. Camps, "Monet: Moments embedding network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3175–3183.
[17] Q. Wang, P. Li, W. Zuo, and L. Zhang, "RAID-G: Robust estimation of approximate infinite dimensional Gaussian with application to material recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4433–4441.
[18] P. Li, J. Xie, Q. Wang, and Z. Gao, "Towards faster training of global covariance pooling networks by iterative matrix square root normalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 947–955.
[19] S. Li, W. Deng, and J. Du, "Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2852–2861.
[20] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon, "Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Nov. 2011, pp. 2106–2112.
[21] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.
[22] Y. Fan, J. C. K. Lam, and V. O. K. Li, "Multi-region ensemble convolutional neural network for facial expression recognition," in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, Sep. 2018, pp. 84–94.
[23] Y. Li, J. Zeng, S. Shan, and X. Chen, "Occlusion aware facial expression recognition using CNN with attention mechanism," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2439–2450, May 2019.
[24] J. Zeng, S. Shan, and X. Chen, "Facial expression recognition with inconsistently annotated datasets," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 222–237.
[25] Z. Yu and C. Zhang, "Image based static facial expression recognition with multiple deep network learning," in *Proc. ACM Int. Conf. Multimodal Interact.*, Nov. 2015, pp. 435–442.
[26] X. Liu, B. V. K. V. Kumar, J. You, and P. Jia, "Adaptive deep metric learning for identity-aware facial expression recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 522–531.
[27] E. D. Cubuk, B. Zoph, and D. Mane, "AutoAugment: Learning augmentation policies from data," 2018, *arXiv:1805.09501*. [Online]. Available: https://arxiv.org/abs/1805.09501
[28] Z. Gao, J. Xie, Q. Wang, and P. Li, "Global second-order pooling convolutional networks," 2018, *arXiv:1811.12006*. [Online]. Available: https://arxiv.org/abs/1811.12006